Why Linux

- World Programmer`s Community Product.
- Open Source Code.
- Stable in relation to Windows.
- Derived from the proven industrial OS Unix.
- More than 85 % Servers use Linux/Unix.
- About 75% mobile devices are running Android (Linux variant).
- Linux can scale. It runs on over 99% of the top 500 supercomputers in the world.
- Redmond project.
- SW: Win <=> Lin <=> Mac OS X

Linux distros

- Ubuntu , Kubuntu , Lubuntu , Debian
- Manjaro , Arch Linux
- OpenSUSE , CentOS
- Fedora , Gentu ,
- Linux Mint
- Kali Linux
- ...

Distros domestically produced

Astra Linux , GosLinux , Альт 8 СП ,
Ось , AlterOS , ROSA Linux ,
Sailfish , ICLinux , Calculate Linux

гарантии безопасности

МСВС

Internet Resources

Ubuntu 20.04 LTS

https://ubuntu.com https://ubuntu.ru

https://askubuntu.com https://askubuntu.ru

https//losst.com https://habr.com

https://ru.stackoverflow.com

https://www.youtube.com (video lectures)

Literature

- Андрей Робачевский. Операционная система UNIX. - С-Пб.: БХВ - Санкт-Петербург, 1999.
- Уильям Стивенс. UNIX: взаимодействие процессов - С-Пб. : Питер, 2002.
- Уильям Стивенс. UNIX: разработка сетевых приложений. – С-Пб.: Питер, 2003.
- Теренс Чан. Системное прогр-ие на C++ для UNIX. - К.: Издательская группа BHV, 1999.
- Дэвид Тейнсли. Linux и Unix: прогр-ие в shell. - К.: Издательская группа BHV, 2001.
- Brown C. Unix Distributed Programming. Prentice Hall International (UK) Limited, 1994.
- Machtelt Garrels. Bash Guide for Beginners (Second Edition). - Fultus Corporation, 2004.
- William E. Shotts, Jr. The Linux® Command Line. A LinuxCommand.org Book, 2009

Literature

- Методичка для лабораторных работ:

  Шмаков В.Э.,

  Открытые системы и Linux-технологии.

  Учебное пособие. Издательство СПбПУ, С-Пб, 2018.

- На библиотечном ресурсе:

  http://elib.spbstu.ru/dl/2/id18-15.pdf/info

Study

- Лекции
- Лабораторные работы

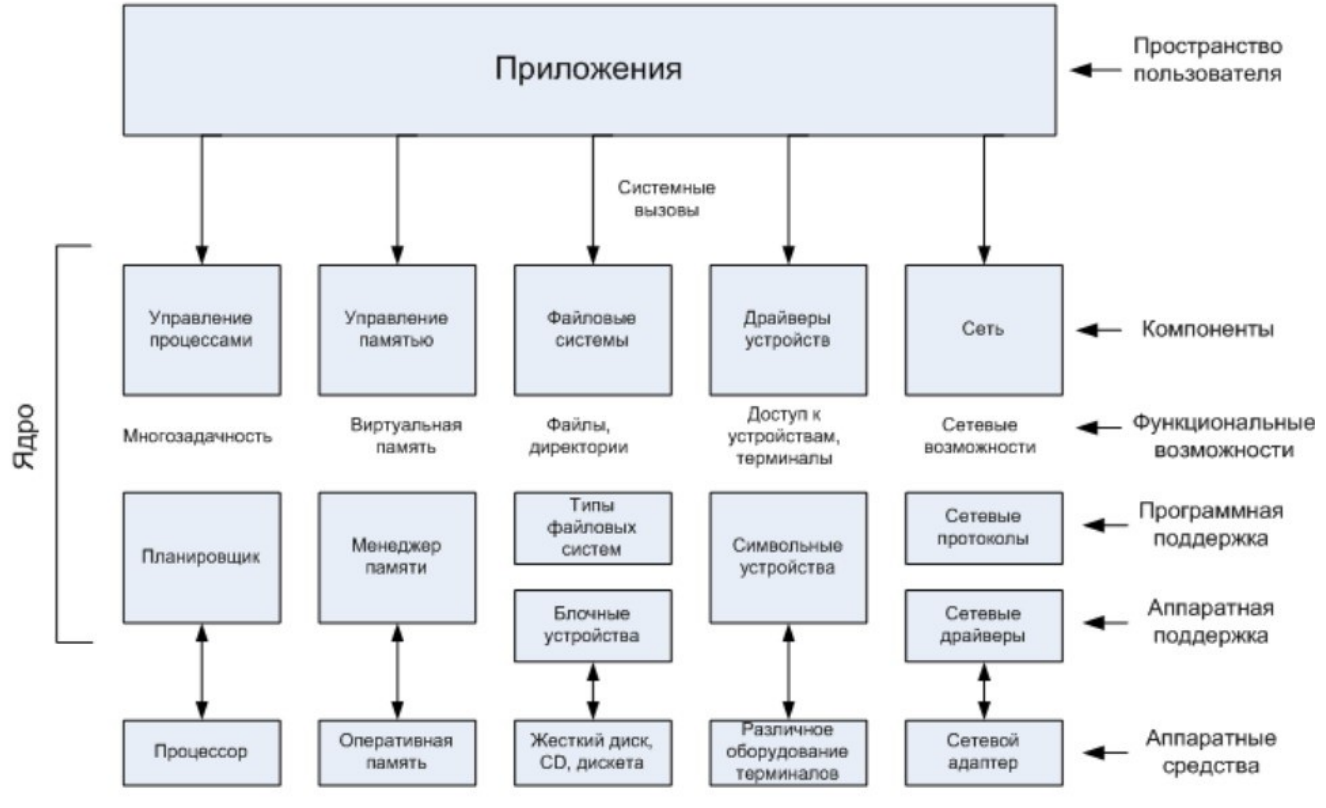
Интегрированный экзамен с оценкой

(отчет по лабораторным)

Laboratory

- Базовые команды ОС, права доступа.
- Запуск и завершение процессов.
- Перенаправление ввода-вывода. Програмные каналы.
- Учетные записи. Переменные окружения.
- Диалоговый и фоновый режимы исполнения процессов.
- Генерация и обработка сигналов.
- Синхронизация процессов семафорами.
- Обмен посредством очередей сообщений.
- Работа с разделяемой памятью.
- Коммуникация на сокетах по сети.

Kernel structure



File System_file types

Типы файлов

В Linux существует **6 типов** файлов,

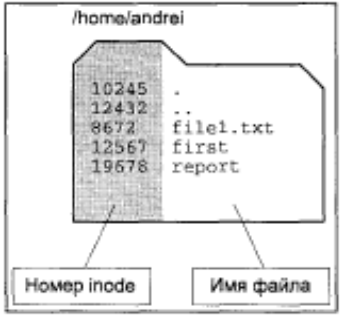
различающихся по функциональному назначению и действиям ОС при выполнении операций над файлами:

- regular file - обычный файл
- directory - каталог
- special device file - специальный файл устройства
- named pipe - очередь FIFO или именованный канал
- link - связь
- socket – сокет

File System_inode

Каждый файл имеет связанные с ним метаданные, хранящиеся в индексных дескрипторах - **inode**, содержащие все характеристики файла.

Каталог - это файл, содержащий имена,
находящихся в нем файлов, и указатели
на их метаданные.



Когда пользователь пытается получить доступ к файлу,
происходит следующее:

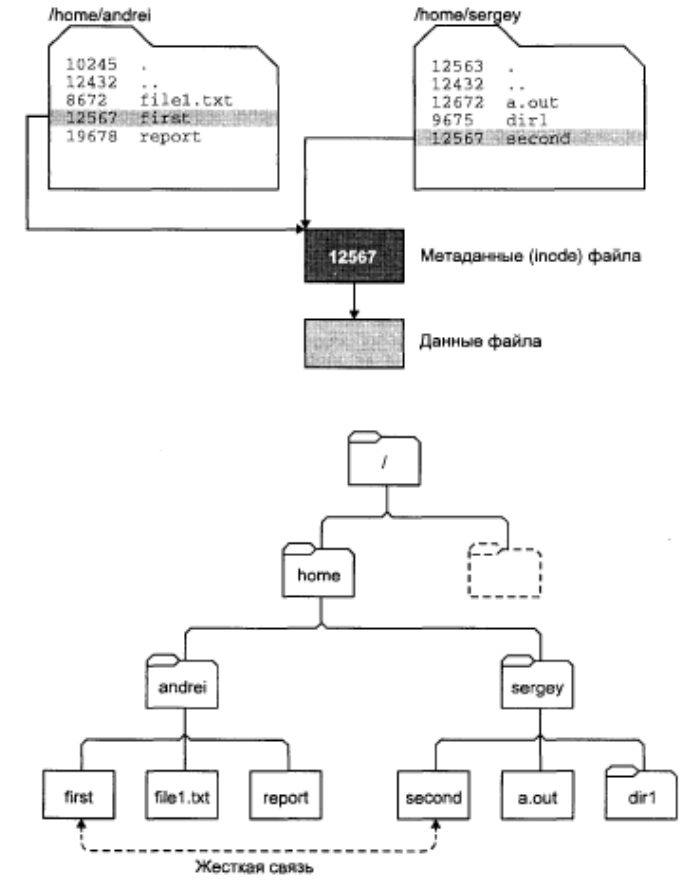
- по имени файла в таблице директориев определяется соответствующий ему номер иноды,
- по номеру иноды происходит обращение к **Inode table** и считываются метаданные,
- начинается работа с файлом, если это возможно.

File System_pipe_block/character

Специальные файлы (special device files) устройств обеспечивают доступ к физическим устройствам.
Различают символьные (character) и блочные (block) файлы устройств, соответствующие разным типам устройств и режимам доступа.
Доступ к устройствам осуществляется посредством открытия, чтения/записи и закрытия специального файла устройства.

Конвейер (FIFO) или именованный канал (named pipe) – это тип файла, используемый для связи между процессами.

File System_link



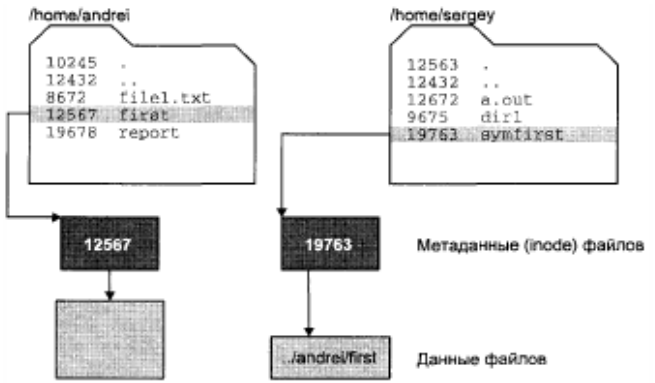
Файл может иметь несколько имен в файловой системе.

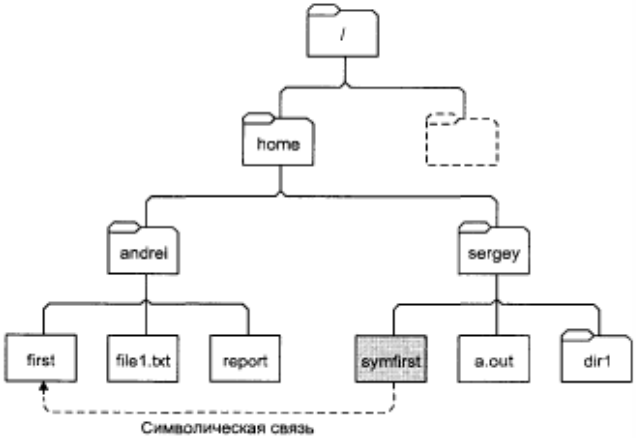
Имена жестко связаны с метаданными, в то время как сам файл существует независимо от того, как его называют в файловой системе.

Обращение по любому из имен.
Все изменения синхронны.

Удаление по одному из имен не приводит к удалению самого файла (остается под другими именами).

File System_link symbolic





Помимо организации жесткой связи возможно установление символической связи (гибкой ссылки) с помощью специального файла, который будет только косвенно адресовать целевой файл.

Данные такого специального файла (являющегося символической ссылкой) содержат только полное имя целевого файла, то есть ссылаются на него.

File System_sockets

Socket <IP Address : Port Number> (Domain, Type, Protocol)

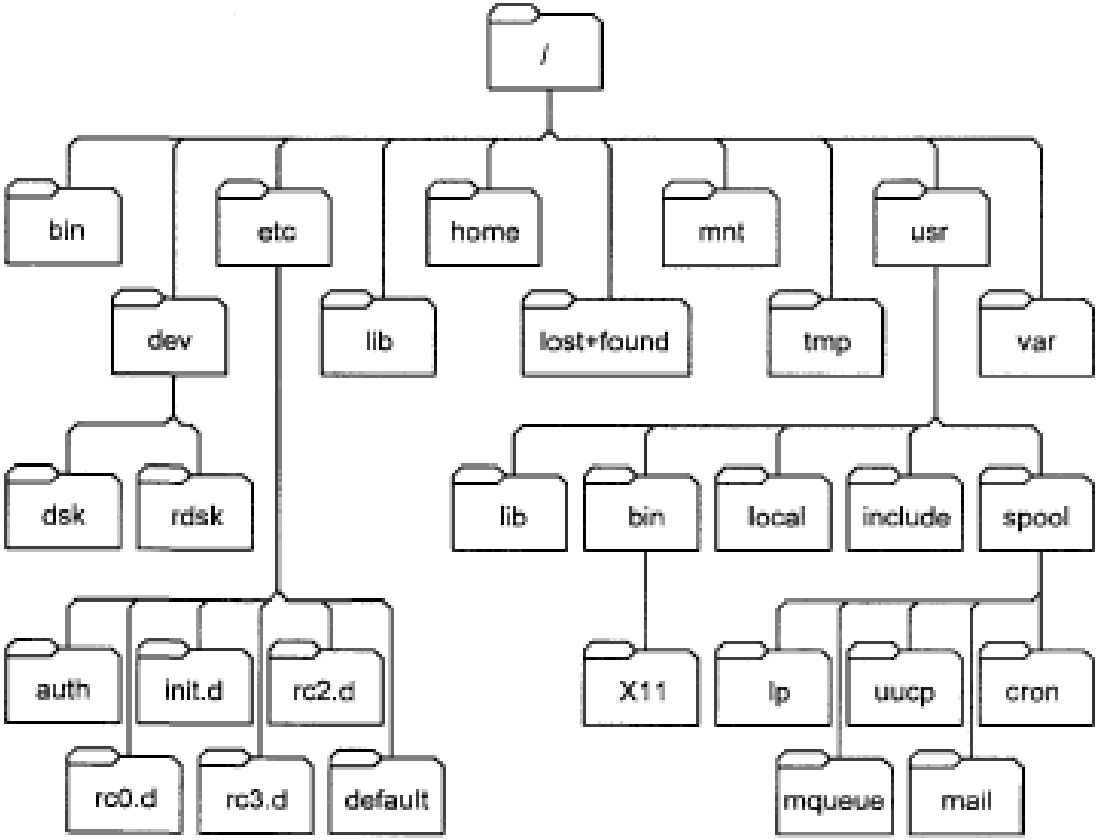
- AF_UNIX Interprocess Communication.
- AF_INET Communication over Network.

- SOCK_STREAM Передача потока данных с предварительной установкой соединения. Надежный канал передачи, фрагменты данных не теряются, не переупорядочиваются и не дублируются.
- SOCK_DGRAM Передача данных в виде отдельных сообщений, без предв. Установки соединения. Ненадежно, но быстее. Допускается передача нескольким получателям (multicasting) и широковещательная рассылка (broadcasting).
- SOCK_RAW Это тип низкоуровневых сокетов. Можно даже формировать заголовки сообщений.

File System_sockets

- Для реализации SOCK_STREAM используется протокол TCP.
- Для реализации SOCK_DGRAM используется протокол UDP.
- Тип SOCK_RAW используется для низкоуровневой работы с протоколами IP, ICMP и др.

File System_structure



File System_structure

Файлы и каталоги располагаются в рамках структуры,
порожденной **root directory**, независимо от их физического местонахождения.

/bin находятся наиболее часто употребляемые команды
и утилиты системы.

/dev содержит специальные файлы устройств,
являющиеся интерфейсом доступа к периферийным устройствам.

/etc в этом каталоге находятся системные конфигурационные файлы, утилиты администрирования, скрипты инициализации системы.

/lib находятся библиотечные файлы языка С и других языков программирования.
Стандартные названия библиотечных файлов имеют вид типа libx.a или libx.so.
Часть библиотечных файлов также находится в каталоге **/usr/lib**.

File System_structure

/lost+found каталог "потерянных" файлов. При аппаратных сбоях могут возникать ошибки целостности файловой системы, что может приводить к появлению «безымянных» файлов.

/mnt стандартный каталог для временного связывания (монтирования) физических файловых систем к корневой для получения единого дерева логической файловой системы.

/usr находятся подкаталоги различных сервисных подсистем, исполняемые файлы утилит UNIX, подкаталоги электр. почты, а также и дополнит. пользовательские программы, библиотеки т. д.

/var используется для хранения врем. файлов разл. сервисных подсистем системы печати и т. д.

/tmp предназначен для хранения временных файлов, необходимых для работы различных подсистем UNIX, а также файлов пользователей (даже для работающих под гостевым аккаунтом **Guest account**).

/home каталог для размещения домашних каталогов пользователей. Например, имя домашнего каталога пользователя **alex** будет называться **/home/alex** .

Command Line_structure

- Запуск консоли (терминала) **Ctrl+Alt+T**
- Формат командной строки:

command options arguments

- Отдельные поля разделяются пробелами или табуляцией
- В качестве arguments указывается объект с которым работает **command**
- В поле options аргументы, модифицирующие команду, обычно начинаются с “-“
- Синтаксис, семантика и аргументы описаны всеобъемлюще в “**man page**” , имеющемся для каждой из команд Linux
- Для вызова системы помощи набираем на консоли:

$ **man** command

$ **man** -k keyword

Commands_Directories

- **pwd** print working directory
- **cd** change working directory
  - no argument changes to **home** directory
  - .. move up one level
  - ~alex changes to home directory of user alex
  - “.” means current directory,
  - “..” means directory up in the tree
- **mkdir** create a directory
- **rmdir** remove a directory
- **ls** list directory contents
  - **ls** -l long listing
  - **ls** -a list of all files (including starting with “.”)

Commands_Files

- **rm** remove (delete) a file
- **cp** copy a file or directory
- **mv** move a file, includes renaming
- **cat** concatenate (list) files and print on the standard output
- **chmod** change permissions (files mode bits)

Use + and – with a single letter

u user (owner of file)
g group
o others
a all (includes user, group and others)

chmod u+w filename
gives user write permission
chmod g+r filename
gives group read permission
chmod a-r filename
ensures no-one can read file

(various notations of **chmod** arguments are possible)

Intput/output redirection

`>` перенаправить стандартный вывод в файл

```
command > outfile
```

`>>` добавить стандартный вывод в файл

```
command >> outfile
```

`<` перенаправить ввод из файла

```
commmand < infile
```

`|` pipe конвейеризация вывода команды на вход другой команды

```
command1 | command2
```

Compressing Archiving

tar *tape archive and retrieval*

объединяет множество файлов в один .tar файл (или извлекает)

`tar -cf textfiles.tar *.tex`

создает `textfiles.tar` файл, содержащий все файлы с расширением `.tex`

`tar -xf textfiles.tar` извлекает содержимое `.tar` файла

Files Naming Conventions

- Использование расширения имени файла в Linux необязательно.
- Можно использовать как одно, так и несколько полей расширений имени.
- Существуют соглашения по присвоению некоторых расширений:

  *.txt , *.odt , *.pdf , *.c , *.cpp и т.д.
  Объектные библиотеки - *.a , *.so
  Исполняемые файлы - *.out , *.o
  Скрытые (hidden) файлы: имена начинаются с точки .*

- Первые два элемента в любом каталоге адресуют:

  сам этот каталог (current) под именем " . "
  и родительский каталог под именем " .. "

- Создание файла на терминале, например:

  $ cat > myfile.txt

Files Owners Permissions

- Файлы в Linux имеют два типа владельцев:
  пользователя (user owner) и
  группу (group owner).
- Группа - определенный список пользователей системы.
  Пользователь системы может быть членом нескольких групп
  одна из которых является первичной (primary),
  остальные - дополнительными (supplementary).
  При этом владелец-пользователь может не являться
  членом группы, владеющей файлом (*гибко*).
- На этапе создания файла его владельцем-пользователем становится тот пользователь, который его создает.
  Определить владельцев файлов можно командой ls -l

```
- rw- r- -  r--  1   alex  dept   246587  Dec 22 19:34  file2.txt
- rw- rwx r--  1   andy  empl   317191  Nov 18 21:04  sm.txt
```

- Изменение владельца файла или прав доступа к файлу
  может осуществлять только владелец файла или суперпользователь **su**.

Files Owners Permissions

- Права доступа к файлу контролируют такие операции, как чтение **r** (**read**), запись **w** (**write**) и запуск на выполнение **x** (**execute**) (для исполняемых файлов).
- В Linux существуют 3 базовых класса доступа, в каждом из которых устанавливаются соответствующие права доступа:
  User access **u** для владельца-пользователя файла,
  Group access **g** для членов группы, являющейся владельцем файла,
  Other access **o** для остальных пользователей.
- Отсутствие прав доступа отображается символом “-” .
- Например, права доступа к файлу a.out :

- rwx r-x r- - 1 andy student 4889 Nov 23 10:15 a.out

Тип файла	**Права владельца-пользователя**	**Права владельца-группы**	**Права остальных пользователей**
обычный	чтение, запись, выполнение	чтение, выполнение	чтение

Directories Permissions

- Право чтения каталога позволяет получить только имена файлов, находящихся в данном каталоге.
  Чтобы получить дополнительную информацию о файлах каталога **ls** -l системе придется "заглянуть" в метаданные файлов, а это уже требует для каталога и права на выполнение.

- Право на выполнения также потребуется для каталога, в который требуется перейти (т. е. сделать его текущим, с помощью команды **cd**).
  Это же право нужно иметь для доступа ко всем каталогам на пути к целевому.

- Особого внимания требует право на запись для каталога.
  Создание и удаление файлов в каталоге требуют изменения его содержимого, и, следовательно, права на запись в этот каталог.
  Самое важное, что при этом не учитываются права доступа для самого файла.
  То есть для того, чтобы удалить некоторый файл из каталога, не обязательно иметь какие-либо права доступа к этому файлу.
  Важно лишь иметь право на запись для каталога, в котором находится файл.

Directories Permissions

Эффекты (права r и x действуют независимо), характерные для Linux. Создание “темных” каталогов, файлы которых доступны только в том случае, если пользователь точно знает их имена, поскольку получение списка файлов таких каталогов запрещено.

*$ **pwd***	Где мы находимся?
/home/andrei	
*$ **mkdir** darkroom*	Создадим каталог
*$ **ls -l***	Получим его атрибуты
d rwx r- - r- - 2 group 65 Dec 22	*19:15 darkroom*
*$ **chmod** a-r+x darkroom*	Превратим его в *"темный"* каталог
*$ **ls -l***	Получим его атрибуты
d -wx --x --x 2 group 65 Dec 22	*19:15 darkroom*
*$ **cp** file_1 darkroom*	Поместим в каталог darkroom некоторый файл
*$ **cd** darkroom*	Перейдем в этот каталог
*$ **ls -l** darkroom*	Попытаемся получить листинг
##permission denied	каталога. Увы...
$ cat file_1	Тем не менее, заранее зная имя
Ok	файла можно работать с ним

(например, прочитать, если есть соответствующее право доступа)

Thanks for your attention

Спасибо за внимание !

vladimir.shmakov.2012@gmail.com